

Especialización en Desarrollo de Software

ST1605 - Procesos Modernos de Desarrollo de Software

TAXONOMÍA (TIPOS) DE SOFTWARE

Introducción

Inspira Crea Transforma

Principio Base

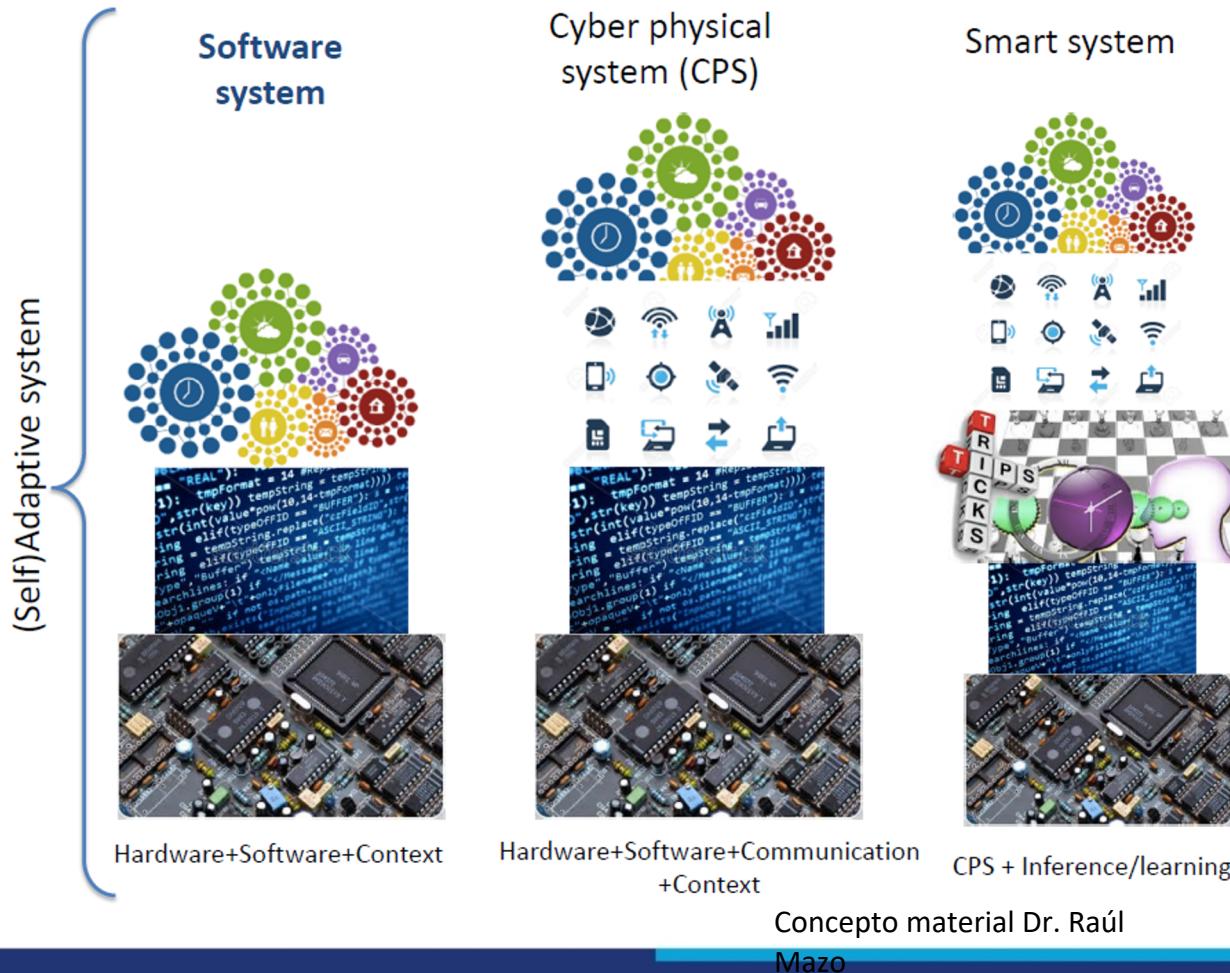
Quienes quieran convertirse en un experto en software, necesitas saber qué es **un sistema de software**.

¡El software no es autosuficiente!

Para adaptarse, necesita contar con (i) hardware para ejecutarse, y con (ii) contexto (interno / externo) para interactuar y crear beneficios.

Concepto material Dr. Raúl
Mazo

Sistemas de Software



Concepto de Software

Definición de Software

En general, el término software se utiliza para describir las **instrucciones digitales y de operación que se contienen en los programas que sirven para guiar a las máquinas** -especialmente ordenadores- en la ejecución de operaciones que deseé (por ejemplo, el procesamiento de datos o que interactúan con periféricos)

Software **se crea a partir de código fuente**, que consiste en secuencias de declaraciones y declaraciones escritas en lenguajes especiales de programación de computadoras.

...

El código fuente se desarrolla por lo general en **uno o más archivos de texto**, un gran conjunto de los cuales puede ser organizados en un árbol de directorios (también llamado árbol de código fuente).

El código fuente es generalmente convertido en un formato digital, un **código ejecutable** de máquina usando un programa especial llamado compilador.

...

Los usuarios finales consiguen generalmente sólo los archivos ejecutables, el código fuente no necesariamente deberá ser revelado al usuario.

Software de código abierto (OSS) incluye o permite fácil acceso al código fuente legible por humanos con el fin de facilitar aún más la modificación o reutilización, sujeto a ciertas condiciones (por ejemplo, licencias).

Dos características clave de software son que es sin cuerpo (**incorpóreo**) y **complementaria a hardware**.

Producto Incorpóreo

El software es un bien no-físico.

Se trata de una secuencia digital de comandos almacenados en soportes físicos, como un disco duro, disco de la memoria o disco compacto.

Software puede grabar y almacenar en forma temporal o forma permanente.

La principal implicación de la forma no-física de software es que, una vez desarrollado, un producto de software puede ser replicado con un costo relativamente bajo y esfuerzo.

El carácter sin cuerpo de software también facilita la transmisión digital (por ejemplo, a través de Internet), lo que puede facilitar y agilizar la entrega.

Complementariedad con el hardware

El hardware proporciona el soporte físico de software.

Del mismo modo, el hardware moderno requiere software para su función y es típicamente incapaz de dar una funcionalidad avanzada en la ausencia de aplicaciones de software.

Como resultado de ello, existen vínculos fuertes y complementariedad entre los mercados de hardware y software.

El rápido ritmo del cambio tecnológico en hardware puede influir desarrollo de software, por ejemplo, los programadores pueden moverse rápidamente para explotar capacidades de expansión en las nuevas generaciones de hardware.

Aproximación a la Taxonomía del Software

¿Qué es una taxonomía?

La ciencia de la **categorización**, o **clasificación**, de las cosas basadas en un sistema predeterminado. En referencia a los sitios web y portales, la taxonomía de un sitio es la forma en que organiza sus datos en **categorías y subcategorías**, que a veces se muestran en un mapa del sitio.

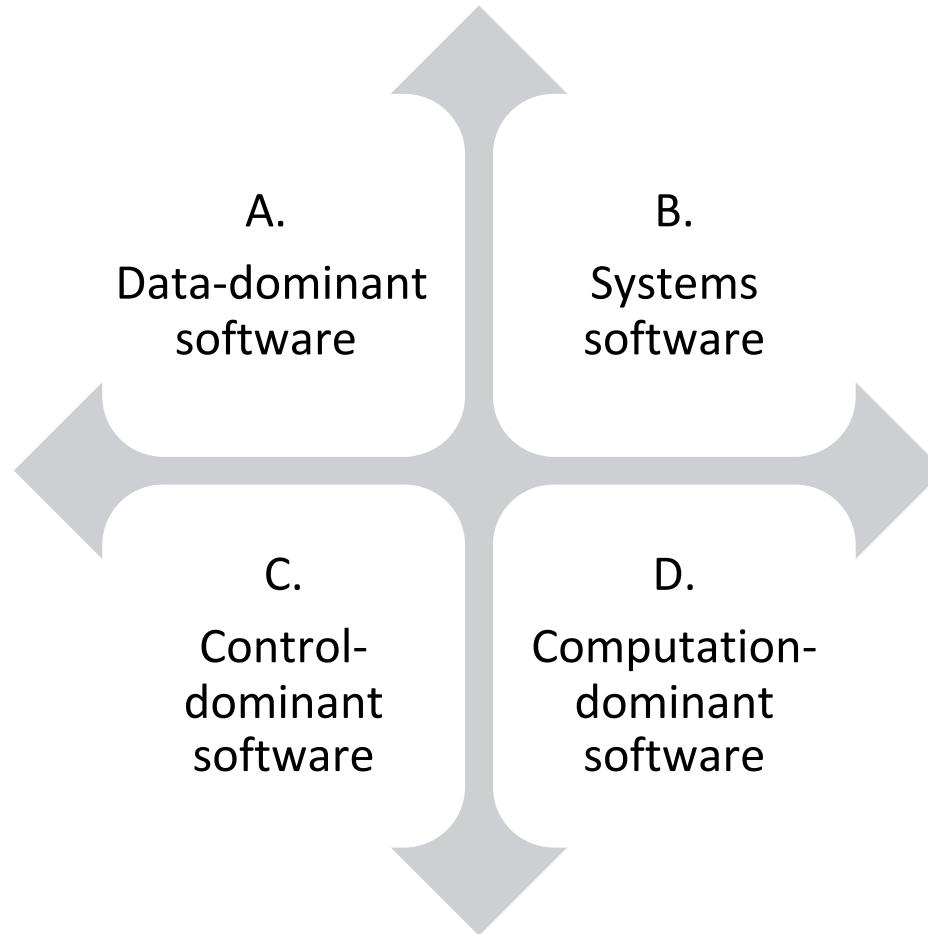
Taxonomías existentes

Es importante tener una buena comprensión de Taxonomías existentes en el campo de la informática.

Antes de crear uno nuevo para aplicaciones de software. Los criterios para la revisión incluyen:

- ¿El propósito y la aplicabilidad de la taxonomía están bien definidos?
- ¿La taxonomía cubre adecuadamente el espacio que nos interesa: tipos de software (incluidos los tipos de programas de aplicación, los tipos de software del sistema y los tipos de componentes de software)?
- ¿Está la taxonomía estructurada de manera simple y sistemática, como para minimizar la herencia múltiple y está utilizando criterios bien definidos para dividir cada categoría en subcategorías?

Aproximación a categorías



Abstracción de las categorías

Sistemas dominantes de datos:

- Cuál es la especialización del usuario objetivo, desde el uso general orientado al consumidor, hasta el software empresarial más específico para los usuarios de ingeniería / tecnología

Software de sistemas:

- Qué nivel de abstracción está logrando, desde la abstracción de hardware de bajo nivel de los sistemas operativos hasta los backplanes de software de alto nivel como Eclipse

...

Software dominante en control:

- Cómo está controlando el sistema, desde pequeños sistemas integrados hasta operaciones extremadamente grandes como un aeropuerto

Software dominante en computación:

- De qué enfoque conceptual es su problema, desde programas prácticos como el programa SETI hasta áreas teóricas como la inteligencia artificial.

A - Data-dominant software

A.con

- Consumer-oriented software

A.bus

- Business-oriented software

A.des

- Design and engineering software

A.inf

- Information display and transaction entry

B - Systems software

B.os • Operating systems

B.net • Networking / Communications

B.dev • Device / Peripheral drivers

B.ut • Support utilities

B.mid • Middleware and system components

B.bp • Software Backplanes (e.g. Eclipse)

B.svr • Servers

B.mal • Malware

C - Control-dominant software

C.hw.

- Hardware control

C.em.

- Embedded software

C.rt.

- Real time control software

C.pc.

- Process control software (i.e. air traffic control, industrial process, nuclear plants)

D - Computation-dominant software

D.or.

- Operations research

D.im.

- Information management and manipulation

D.art.

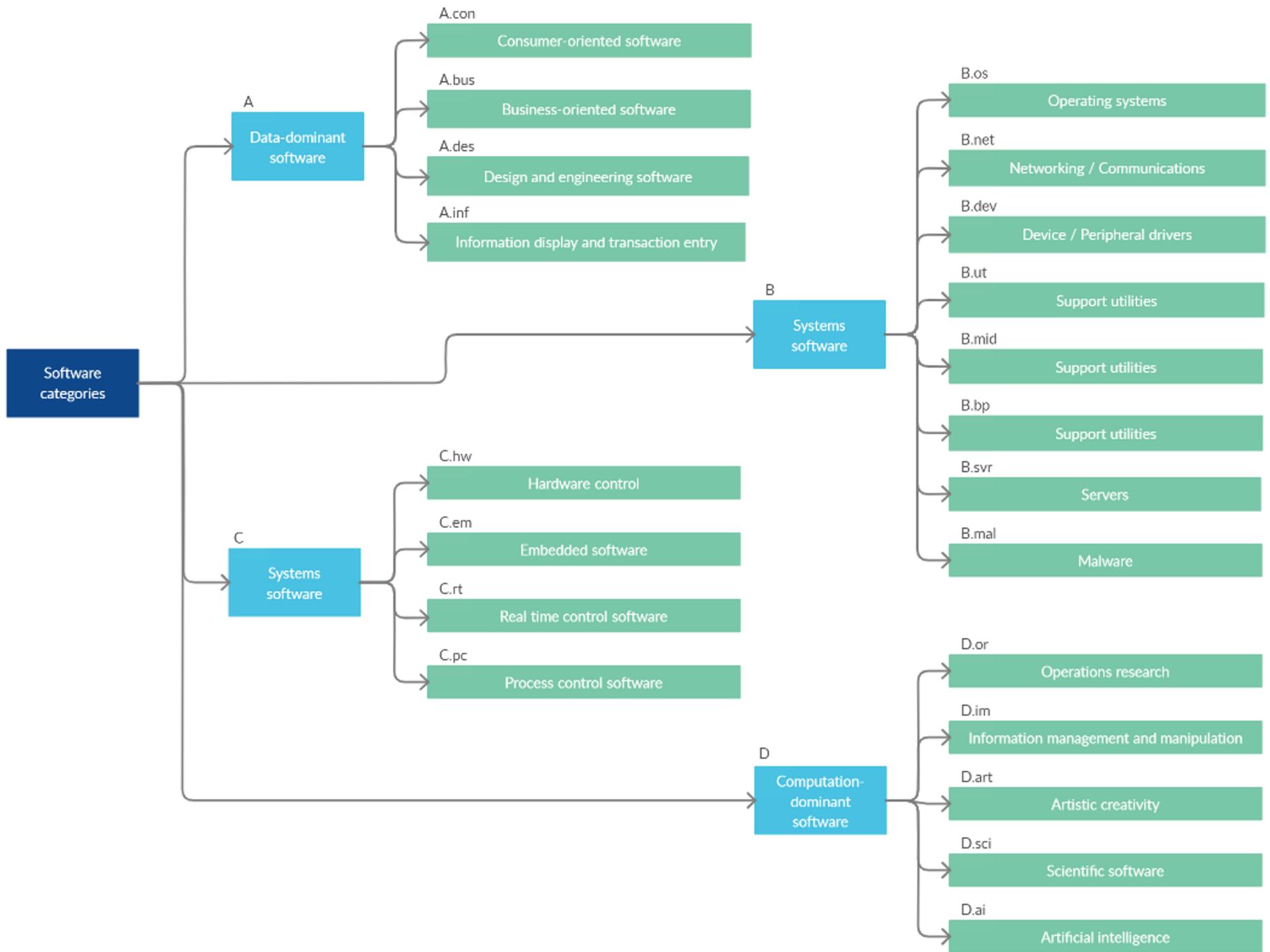
- Artistic creativity

D.sci

- Scientific software

D.ai

- Artificial intelligence



Utilidad de las taxonomías

Basado en la evidencia en la literatura que relaciona varias herramientas, técnicas y métodos con categorías de aplicaciones específicas, la taxonomía del software ayudará a **guiar a los profesionales del software a la hora de tomar decisiones** sobre elementos como los siguientes:

...

Metodologías de software.

Patrones de diseño y marcos de trabajo de componentes (frameworks).

Algoritmos.

Herramientas de ingeniería de software como IDE

Datos y formatos de intercambio como XML, YAML, y JSON.

Arquitecturas y estilos arquitectónicos.

Técnicas de pruebas incluidas las de unidad, integración, sistema, aceptación del usuario y pruebas de humo.

Bibliografía y Referencias

- 1 • Material de estudio para curso de Sistemas autoadaptables del Dr. Raúl Mazo, profesor asociado a la Universidad EAFIT, profesor titular Sorbonne University.
- 2 • https://www.if4it.com/SYNTHESIZED/FRAMEWORKS/TAXONOMY/management_software_taxonomy.html
- 3 • Forward, Andrew and Lethbridge , Timothy. A taxonomy of software types to facilitate search and evidence-based software engineering. DOI: 10.1145/1463788.1463807 · Source: DBLP. January 2008

Conformación de equipos

1. https://docs.google.com/spreadsheets/d/1a-SCWHT6qymcYLChwYi_Y7X1087yakd-EHvSMARK-lo/edit?usp=sharing
2. Selección de dominio y enfoque

Taller en clase:

Realizar un diseño de un modelo de proceso de cómo trabajan hoy y basado en lo que hemos hablado de taxonomía de desarrollo el tema que ha seleccionado.

Fin de Tema

Procesos Modernos de Desarrollo de Software
TAXONOMÍA TIPOS DE SOFTWARE